

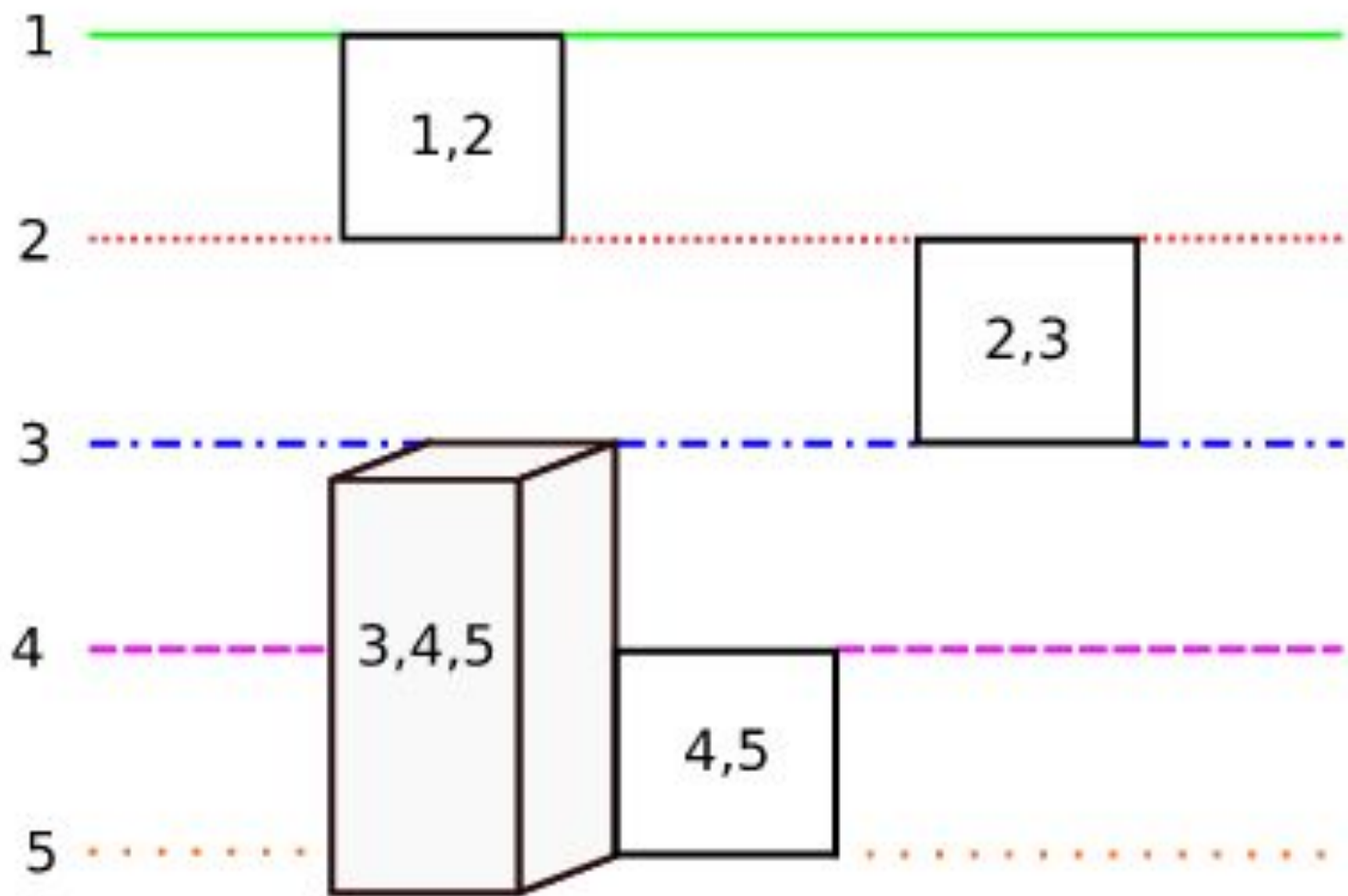
# Matching in *MAPF* met $M^*$

...

Presentatie van het onderzoeksplan

# probleemdefinitie

- Multi agent pathfinding
- Matching
- $M^*$ 
  - “fancy independence detection”
  - plan routes onafhankelijk volgens individueel optimale route
  - houd bij op welke plekken met elkaar in conflict komen
  - Stop met het plannen van routes met zulke conflicten
  - Voeg met verzamelde informatie over eerdere conflicten, nieuwe stappen aan de zoekwachtrij toe die om het conflict heen proberen werken
  - Alsof agents alleen rond conflicten gezamenlijk gepland worden



# Onderzoeksvragen

- Kan  $M^*$  aangepast worden om op een efficiënte manier optimale oplossingen voor  $MAPFM$  te vinden?
  - Hoe kan  $M^*$  aangepast worden om ook matching te doen.
  - Zijn oplossingen in dit geval nog optimaal?
  - Zijn er meer manieren op matching te doen? Zijn deze allemaal optimaal en welke is het snelst?
  - Hoe vergelijkt de rekentijd van  $M^*MAPFM$  tot de rekentijd van andere basisalgoritmen die matching toevoegen?
  - Zijn er manieren om met heuristics specifiek tot  $MAPFM$  betere resultaten te verkrijgen?
  - Kan  $M^*$  en  $M^*MAPFM$  aangepast worden om operator decomposition te doen?
  - Wat gebeurt er als de hoeveelheid agents niet gelijk is aan de hoeveelheid doelposities?

# Matching toevoegen

- Maak iedere staat waarin iedere agent op een eindpunt van de juiste kleur staat een eindstaat (dit kunnen dus meerdere staten zijn)
  - Agents volgen individueel optimale paden, niet rekening houdend met andere agents.
  - Dit optimale pad verandert, als een andere agent van dezelfde kleur op een doel komt dat optimaal is voor deze agent, dan wordt het optimale pad verkeerd.
- Vind iedere mogelijke matching, en draai  $M^*$  daarop
  - Erg traag
  -
- Network flow?
  - Lastig om toe te passen op algoritmen zoals  $A^*$  en  $M^*$